DAVID LEIGH
(08)8332-2627

CONTROL DATA® *Aerospace* Research Department

449 COMPUTER

REFERENCE MANUAL

OCTOBER 1967

CONTROL DATA
CORPORATION

CONTROL DATA AEROSPACE RESEARCH
449 COMPUTER REFERENCE MANUAL

Second Edition

This manual describes the CONTROL DATA, AEROSPACE RESEARCH, 449 miniature computer. This information is preliminary and subject to change without notice.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CONTROL DATA
AEROSPACE RESEARCH
449 COMPUTER

I. INTRODUCTION

The Control Data 449 is a miniature, stored program, general purpose, digital computer designed for use where size, weight, power, and reliability are at a premium. The advanced design techniques used provide moderately fast solutions to complex data processing problems. High reliability, small size and low power are accomplished through the use of off-the-shelf components in a minimum component and yet respectable computer organization. Integrated circuits, multilayer etched circuit boards, and thin film components are used where they show a clear cut advantage in achieving the computer goals. Special integrated circuits have been avoided in an effort to keep small quantity reliability high and costs low. Table 1 is a summary of the Control Data 449 central processor characteristics. The characteristics of a typical computer system employing the 449 central processor are given in Table 2. This system (for which the 449 was originally developed) is a portable, self-powered navigation system employing either a sextant, stadimeter, or a star occultation photometer as a sensor. The tables list some of the expected characteristics of the 449 as constructed for aerospace applications. The engineering model, shown in the photographs, will be identical except that the program memory will contain only 4096 words instead of the 7680 words for which space is provided, the scratch pad consists of 128 24-bit words rather than 256 and the batteries are nickel cadmium (rechargeable) rather than the one shot mercury zinc which would be used where recharging is not practical.

The program memory in the prototype 449 computer is an electrically alterable, nondestructive read memory. Program changes are made by the connection of external ground equipment. The weight and power required by the 449 computer as listed in Table 1 are for the central processor and memory only.

TABLE 1

BASIC COMPUTER CHARACTERISTICS

EXTERNAL

O VOLUME	- 4" CUBE
O WEIGHT	- 4 POUNDS
O POWER	- 4 WATTS PEAK

COMPUTATION

- O PARALLEL ARITHMETIC
- O 24 BIT WORD LENGTH
- O 37 INSTRUCTIONS
- O 2 INDEX REGISTERS
- O 8 WORD REGISTER FILE
- O 28 MICROSECOND ADD TIME
- O 604 MICROSECOND MULTIPLY TIME
- O DOUBLE PRECISION CAPABILITY
- O SUBROUTINE CAPABILITY

MEMORY

NONDESTRUCTIVE READOUT BIAX

- O 7680 INSTRUCTIONS (12 BITS)
- O ELECTRICALLY ALTERABLE BY EXTERNAL MEMORY LOADING EQUIPMENT

DESTRUCTIVE READOUT THIN FILM

- O 256 OPERANDS (24 BITS)
- O SCRATCH PAD····· ALTERABLE BY PROGRAM
- O RANDOM ACCESS

INPUT/OUTPUT

- O 12 BIT PARALLEL PATH TRANSFERS
- O 16 CHANNELS POSSIBLE
- O INTERRUPT CAPABILITY

RELIABILITY

- O MONOLITHIC INTEGRATED CIRCUITS
- O LOW PARTS COUNT
- O RUGGED DESIGN
- O NO MECHANICAL CONNECTORS
- O NEGLIGIBLE INTERNAL HEATING

TABLE 2

COMPUTER SYSTEM CHARACTERISTICS

EXTERNAL

O VOLUME	- 4"x 4"x 9"
O WEIGHT	- 12 POUNDS WITH BATTERIES
O POWER	- 4 WATTS + 5 WATTS FOR 3 SECONDS DURING 7 DIGIT DISPLAY CHANGE

SELF CONTAINED

- O INTEGRAL KEYBOARD
- O INTEGRAL DECIMAL DISPLAY
- O REAL TIME CLOCK - 1 PART IN 10^7 ACCURACY
  SEPARATE POWER SWITCH
- O BATTERY POWER - MERCURY ZINC ONE SHOT - 2 WEEKS
  OF CLOCK OPERATION PLUS 24 HOURS OF COMPUTATION TIME.

II. PHYSICAL DESCRIPTION

The Control Data 449 is a rugged, medium performance, general purpose, 24-bit, digital computer weighing four pounds, yet containing 7680 instruction words in a volume of 64 cubic inches. High reliability is achieved through the use of readily available, high volume produced components in a minimal component configuration, electrically alterable nondestructive readout program memory and absence of internal connectors.

The Control Data 449 consists of the following physical sections:

1) Memory Stack
2) Memory Sense and Digit Unit
3) Plane Select Module
4) Word Select Module
5) Arithmetic and Control Unit.

Additional sections of the 449 system are:

6) Input/Output Unit
7) Keyboard and Display Panel
8) Battery Unit
9) Computer Case.

The sections of the 449 central processor and memory are shown in Figure 1. Figure 2 shows the completed 449 computer system. Figure 3 is a face-down view of the system with the bottom removed, showing central processor and battery unit. The central processor is in the foreground with the memory load connectors (front) and the maintenance connector(side) showing. Figure 4 depicts the arrangement of the physical sections of the computer within the case.

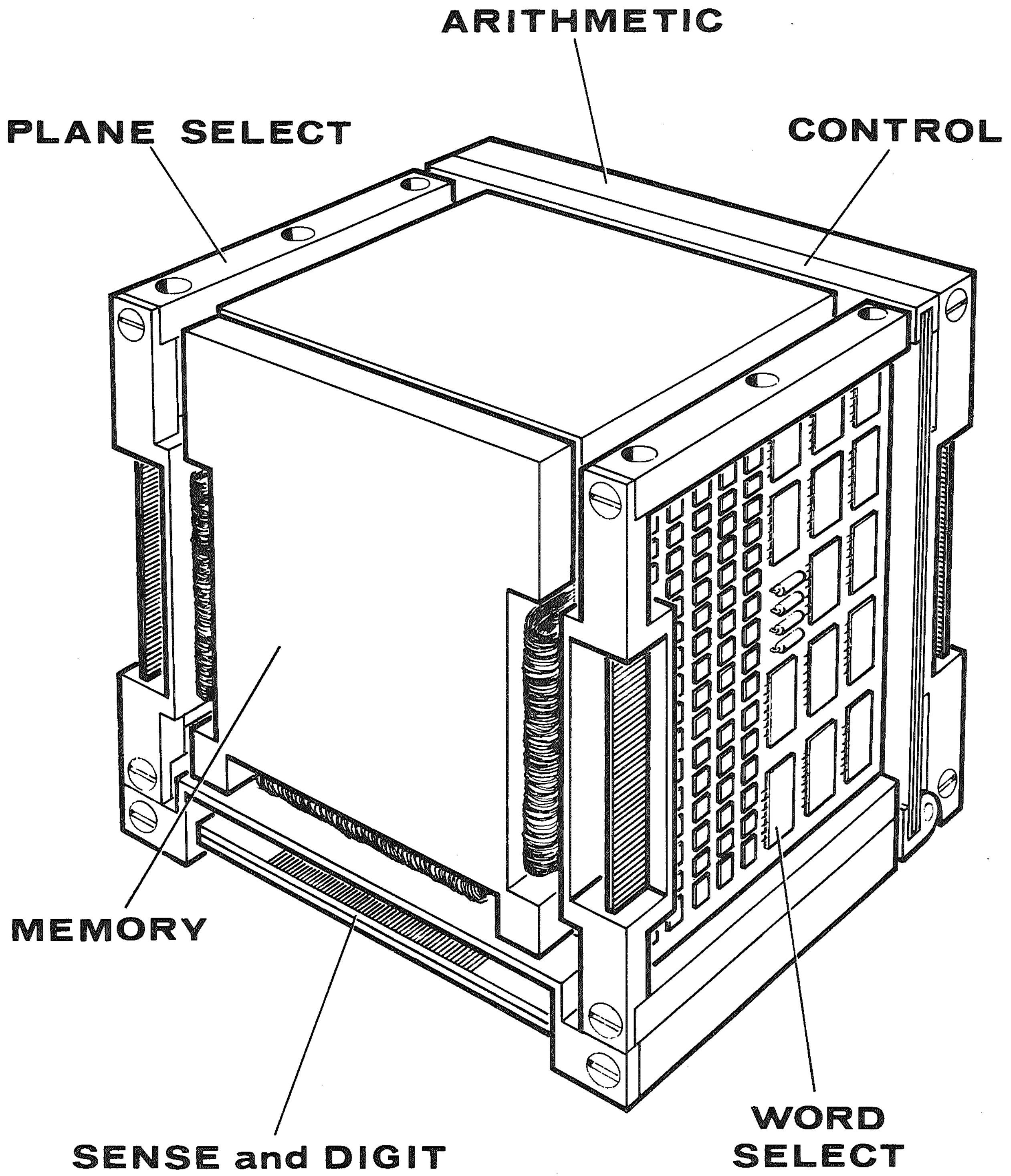


Figure 1: 449 Central Computer

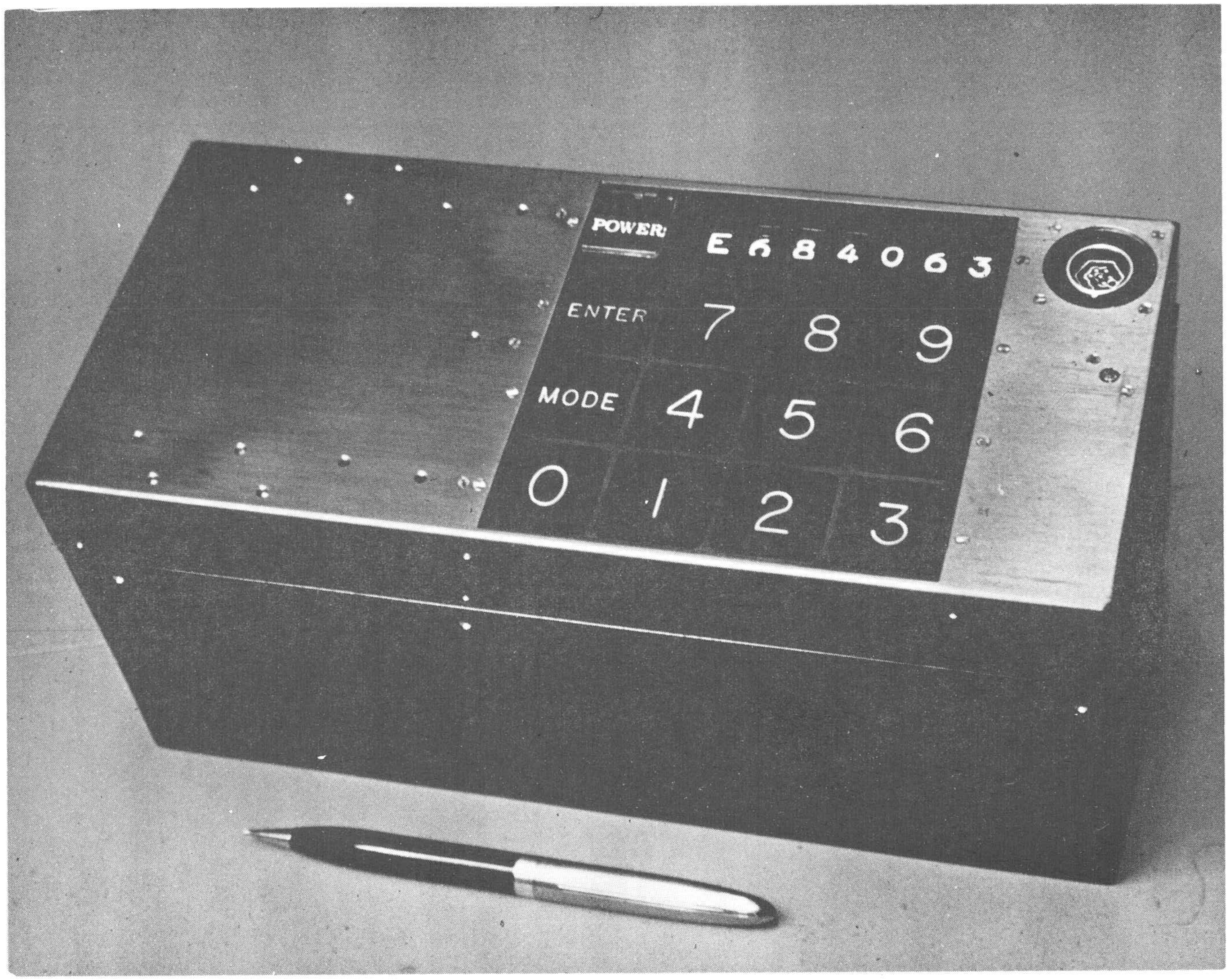


Figure 2: 449 Computer System

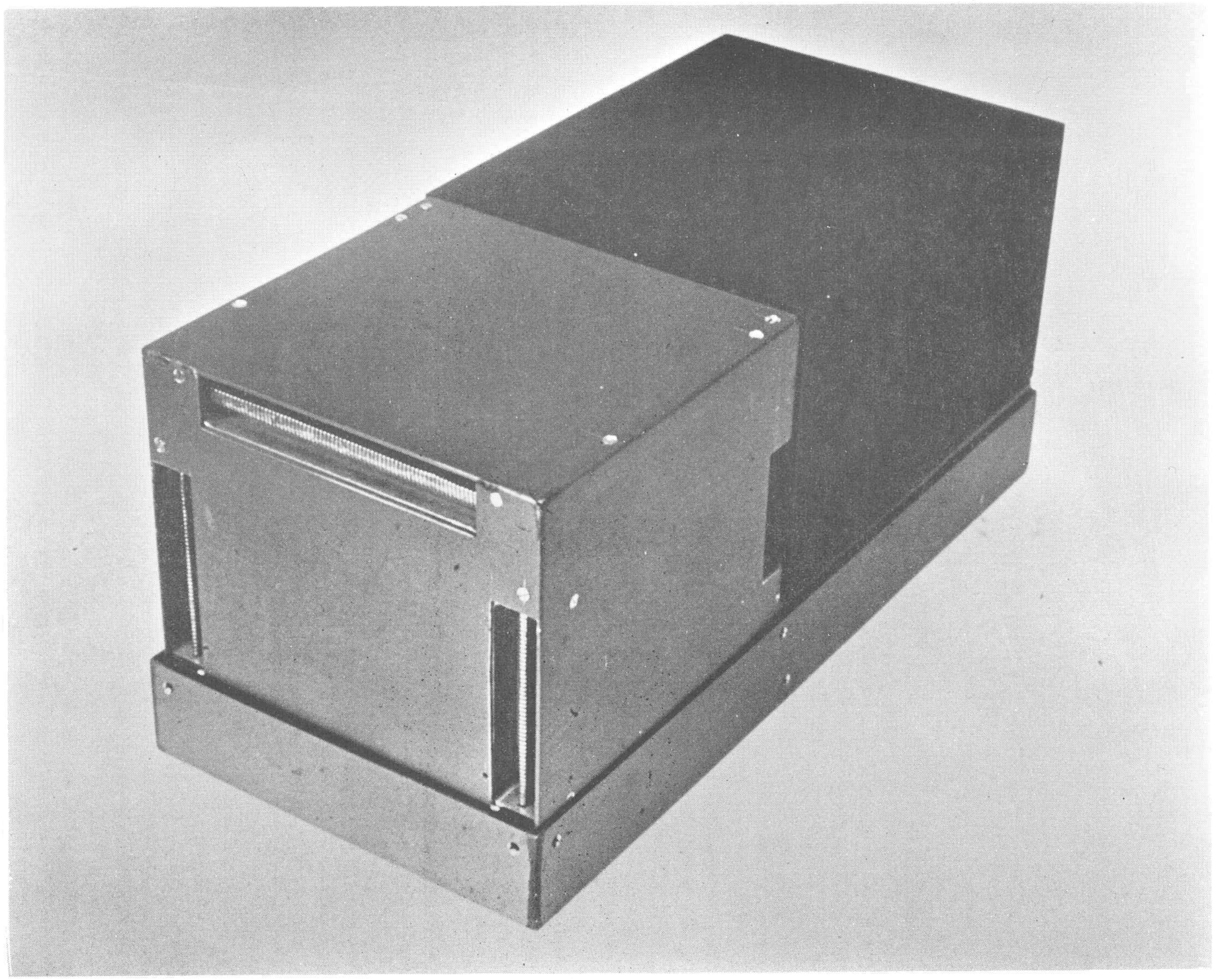


Figure 3: 449 System (Outer Cover Removed)

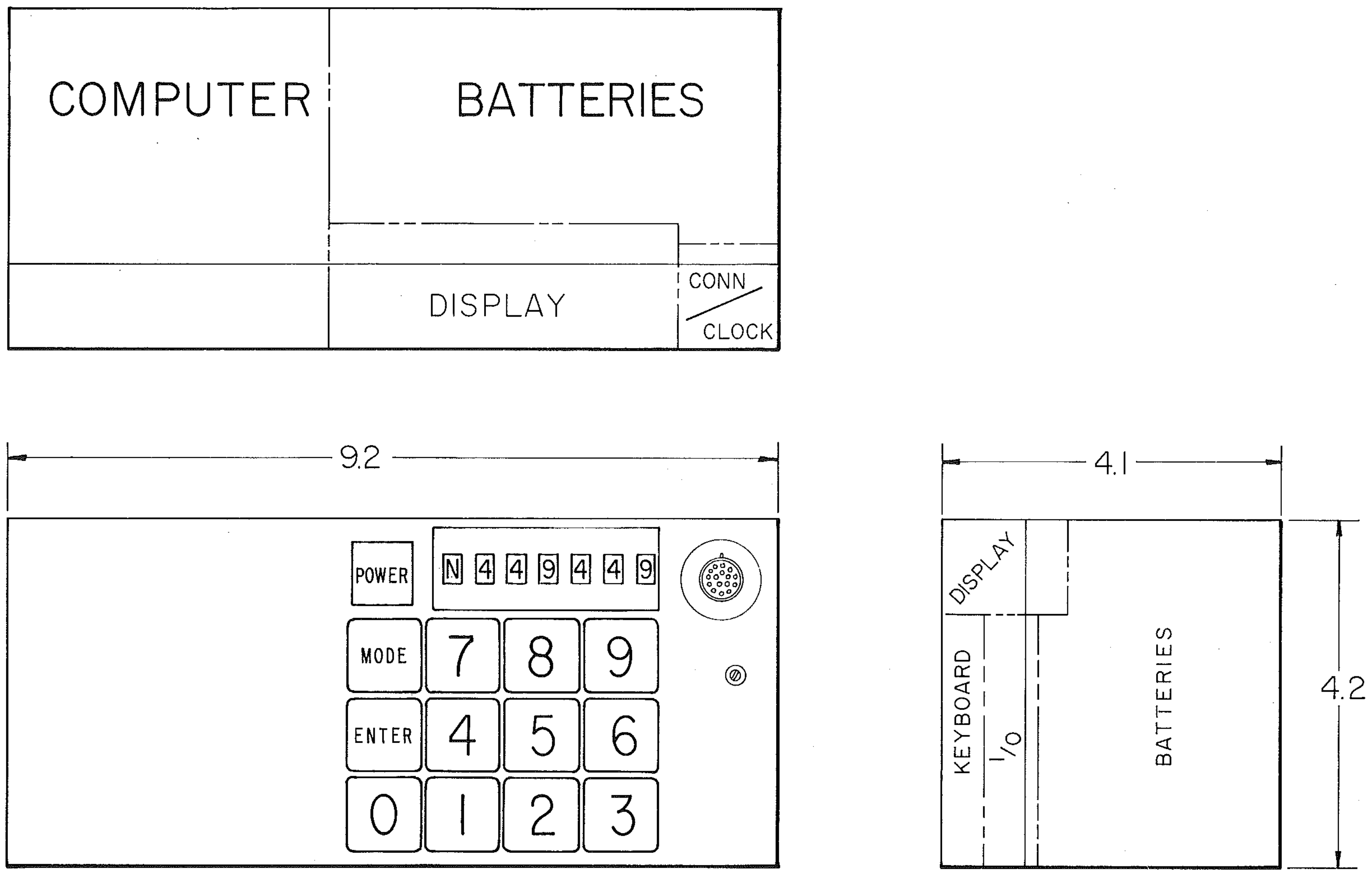


Figure 4: Mechanical Details of 449 System

Memory Stack

The memory stack consists of two sections. They are the 7680 words of Micro Biax* NDRO program memory and the 256 words of DRO film scratch pad. The NDRO section of the memory consists of 15 planes of biax elements, each containing 64 48-bit words and their associated word selection diodes. Stacked directly above it are 8 planes of thin film scratch pad, each containing 64 12-bit words and their diodes. The entire memory stack is contained within a Conetic* shield for magnetic field immunity. Communication with the sense amplifiers is with twisted pairs between the inner and outer shield cams. A photograph of the completed stack is shown in Figure 5.

Memory Sense and Digit Unit

There are a total of 60 sense amplifiers associated with the two memories. Twenty-four are mounted on each of two multilayer printed circuit boards. A third multilayer board provides connections for the remaining 12 sense amplifiers associated with the film memory and the 12 digit drivers. The three boards are stacked in a frame and the assembly installed directly under the memory stack. The two multilayer boards supporting the NDRO sense amplifiers are extended beyond the frame and serve as a connector for the connection of external digit drivers while the program is being loaded.

Word and Plane Select

Word and plane select modules consist of a single multilayer board, each with the associated components located on one side. They are installed in module frames and mounted either side of the stack, thus minimizing interconnection wire length. Again the multilayer boards are continued beyond the frames to provide connectors per word selection while program memory is being loaded.

---

* Raytheon

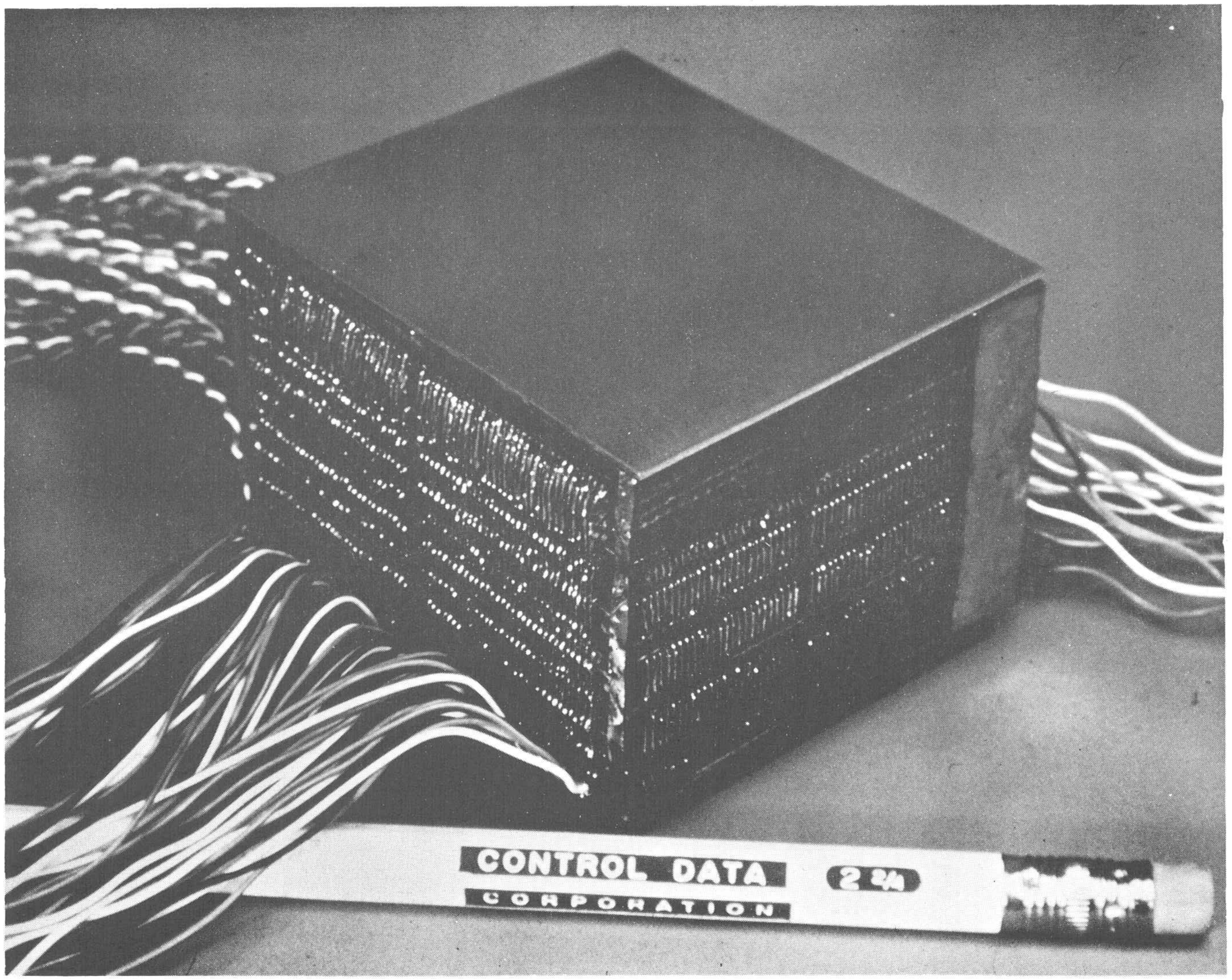


Figure 5: Completed Memory Stack

Arithmetic and Control

Arithmetic and control modules each consist of two multilayer boards permanently mounted back-to-back with integrated circuits on the outer sides. Each side of a module holds approximately 50 circuits and the two modules thus contain the 177 integrated circuits of the central processor. One partially assembled logic module is shown in Figure 6.

Input/Output

The input and output for the 449 system consists of two additional modules, each containing one multilayer card, mounted directly beneath the keyboard and display assembly. The countdown associated with the 23-day clock makes up one module while the other contains the input/output associated with the keyboard, display, and the interface with the programmer's console. A partially assembled I/O module is shown in Figure 7. The large transistors are the wheel drivers.

Battery Unit

The battery for the 449 system is made up of a number of mercury-zinc cells. There are two basic cell sizes for the higher and low current drains. The cells are connected together in a series parallel combination and the entire unit potted to form one assembly.

While the present 449 system will use mercury-zinc nonrechargeable cells, nickel-cadmium rechargeable batteries can be installed in the same manner. All of the taps provided for the computer power are brought out to the front connector for testing or for charging. The entire battery string can be simultaneously charged with a single voltage charger.

Note that some applications of the 449 will not require batteries. It may be advantageous in some cases, however, to use the batteries rather than installing a power supply and using whatever power source is available

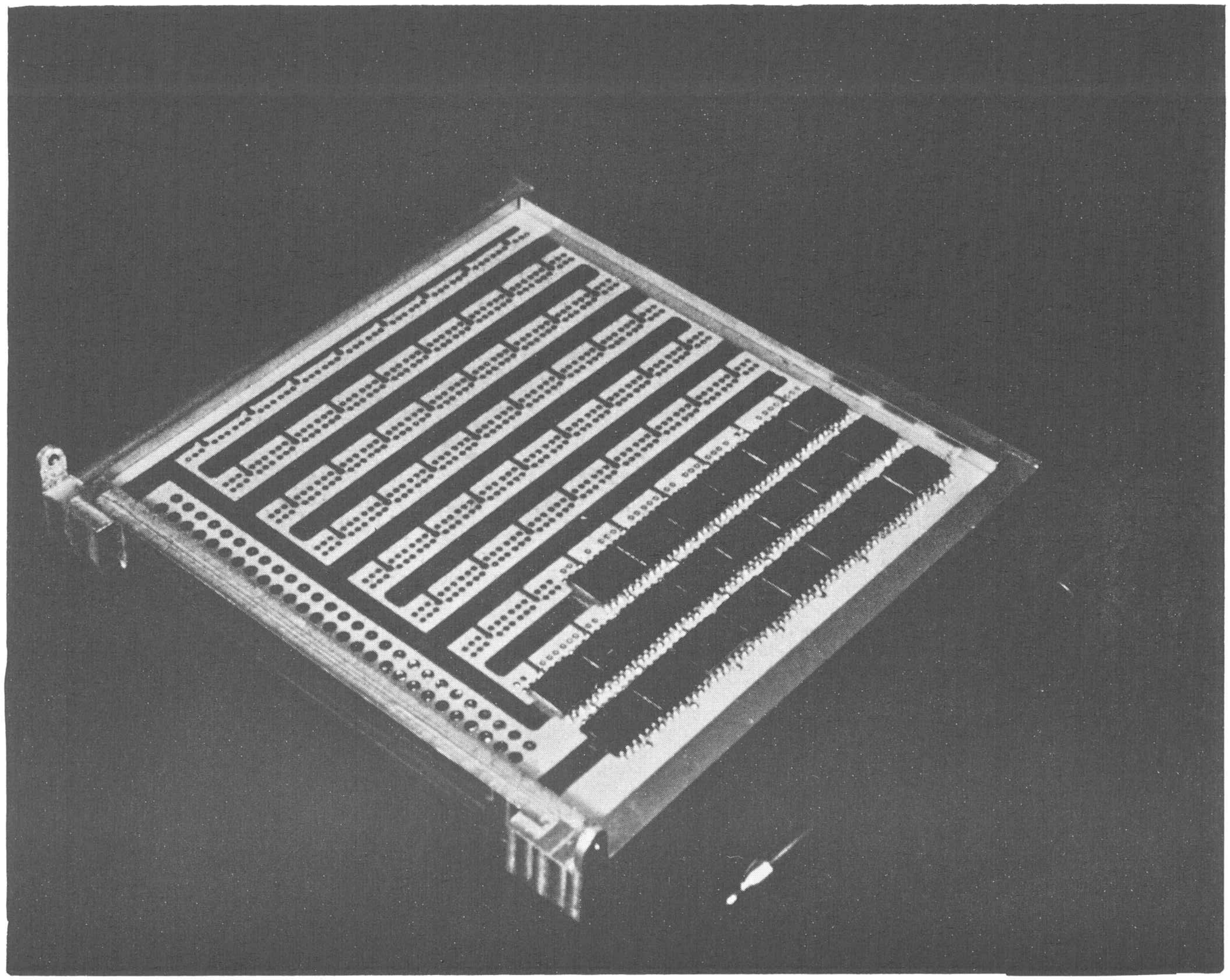


Figure 6: Logic Module

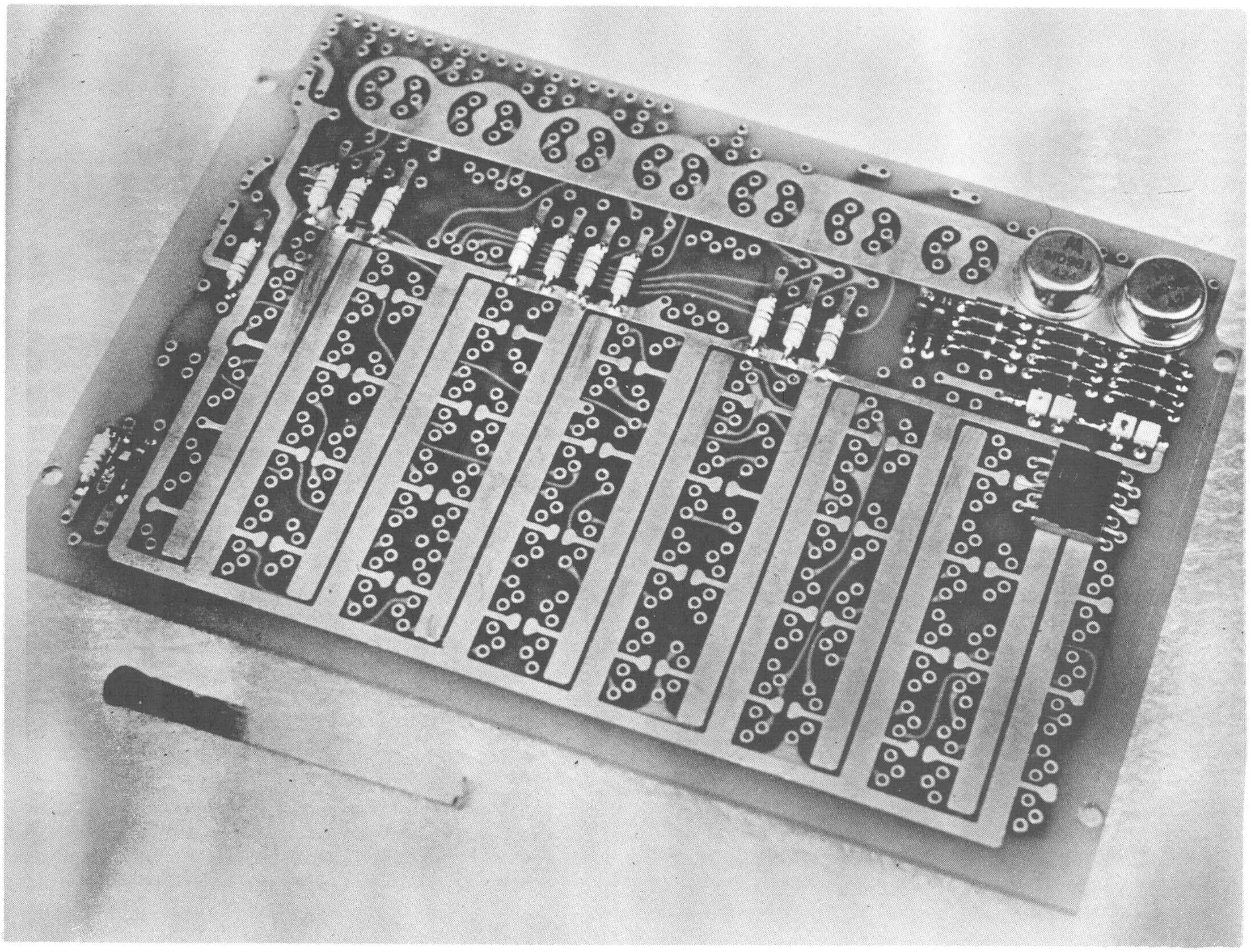


Figure 7: Input/Output Module

to keep the batteries charged. The batteries would, in these cases, probably be considerably smaller. The battery supply can be replaced with a power converter having a volume about 1" x 4" x 4".

Computer Case

The computer front panel is the main structural element of the 449 system. All assemblies with the exception of the battery unit are mounted to the panel. Two covers are provided to complete the assembly. The first or inner cover is a dust cover for all the assemblies except the battery. The outer cover completes the assembly. To load memory or to change batteries, the outer cover is removed. The inner cover then serves as protection while this form of maintenance is being performed.

III. FUNCTIONAL DESCRIPTION

The 449 computer can be divided into sections from a physical and a functional viewpoint as follows:

1) Memory
2) Arithmetic and Control
3) Input/Output

A functional block diagram of the control processor is shown in Figure 8. The I/O for the 449 computer system is shown in Figure 9. This section describes the organization of the computer with respect to memory addressing, instruction format, and word length. The 449 I/O system is described briefly.

A. Memory Section

The 449 computer memory is divided into two sections. The instruction-constant section, containing 1920 memory words of 48-bits each, is implemented with Micro Biax elements. It is electrically alterable and has random access nondestructive readout. Memory loading is accomplished by plugging the computer into the memory load console, which supplies the write and digit currents necessary for memory loading. The loading can be performed as either an inspect and change process, or under paper tape control. A paper tape reader is included as part of the console. While the memory word length is 48 bits, only 12 bits are held in storage on readout. Low order bits of the address determine which set of 12 sense amplifiers are turned on.

The scratch pad consists of 512 words of 12-bits each, and is implemented with thin film elements. The scratch pad is random access and is destructive readout. Both memories operate at a 4 microsecond cycle time.

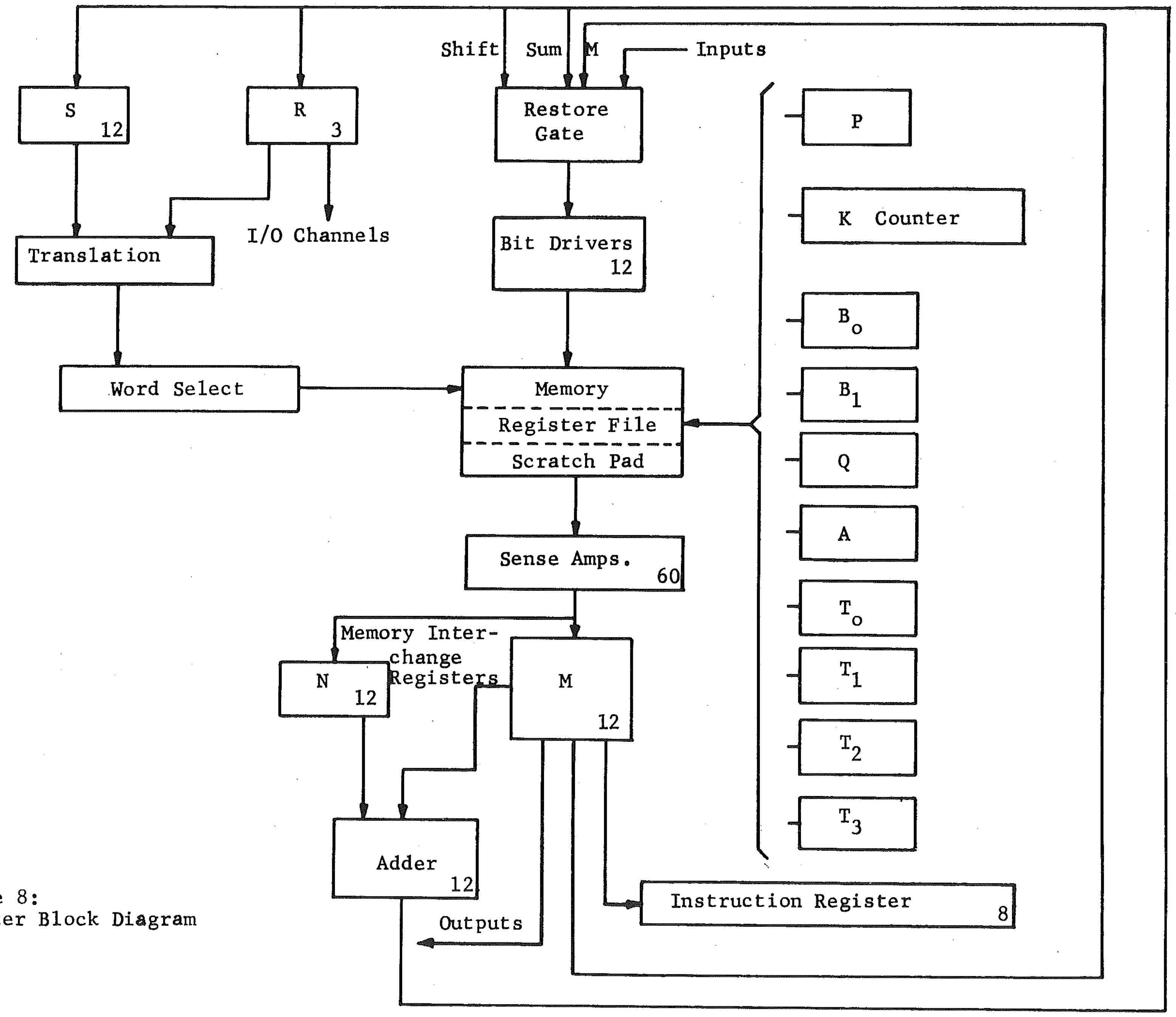


Figure 8:
Computer Block Diagram

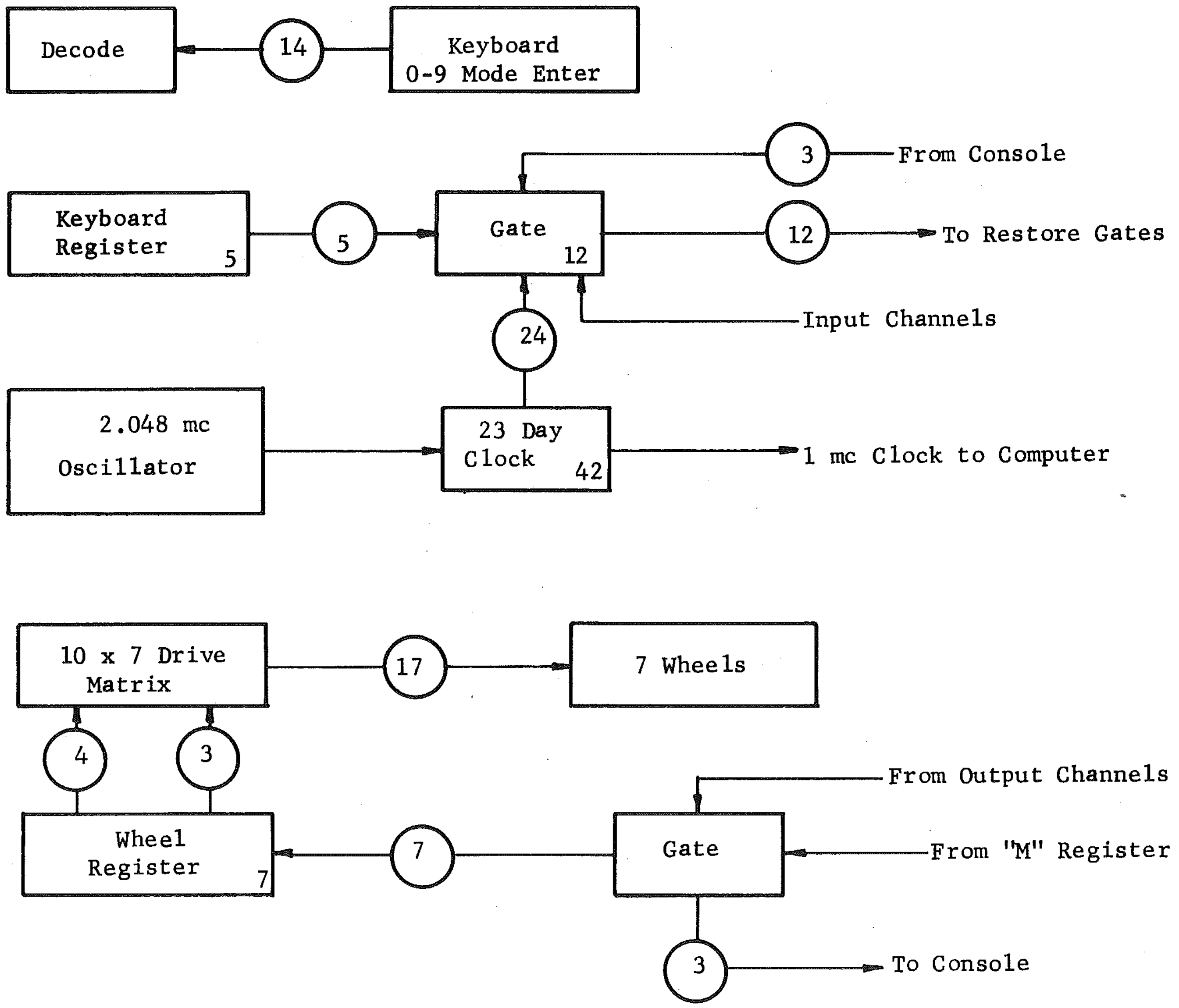


Figure 9: 449 System Input/Output Diagram

B. Arithmetic and Control

Logical Organization

All operational registers are in the memory in an eight word register file. The logic section requires only two temporary holding registers, S and M. Arithmetic results are written directly into memory.

REGISTER FILE

Address	Name	Use
0000L	P	Program Address Counter
0000U	K	Multiply Iteration Counter
0001L	B^1	Index Register
0001U	B^0	Index Register
0002	Q	Product Register
0003	A	Accumulator
0004	T^0	Storage
0005	T^1	Storage
0006	T^2	Storage
0007	T^3	Storage

The basic addressable word length is 24 bits. One operand or two instructions are stored in one memory word. The logic operates in a 12-bit parallel mode. Arithmetic operations are done in two successive 12-bit steps. All arithmetic operations are performed in two's complement.

Instruction Word Format

Twelve bits specify an instruction; therefore, a computer word will hold two instructions, termed upper and lower (Figure 10).

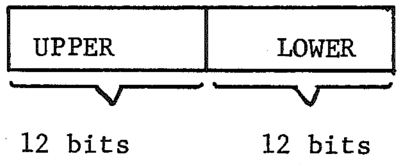


Figure 10: Instruction Location

Instructions are grouped into three general types.

Type 1

f	b	m
3	1	8

Type 2

f	s	y
3	2	7

Type 3

f	s	j	i
3	3	3	3

f	= function code
b	= index register designator
m	= base execution address
s	= sub-function code
y	= operand (sign bit extended)
i	= address of a file register
j	= address of a file register
R	= a 24-bit file register
R_U	= upper 12 bits of a file register
R_L	= lower 12 bits of a file register
A	= R^3
Q	= R^2
P	= R_L^0
K	= R_U^0
B^1	= R_L^1
B^0	= R_U^1

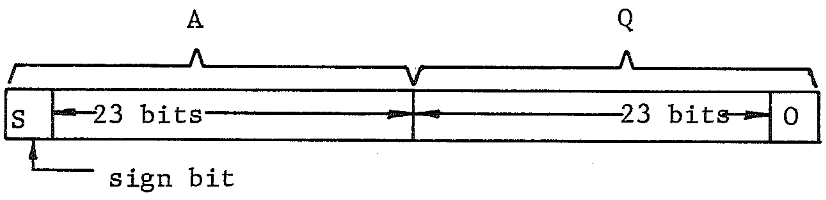
Note. One of the two index registers is always used.

Type 1 Instructions

The effective execution address for all Type 1 instructions is $M = m + (B^b)$ Modulus 2^{12}. Parentheses imply "the contents of" the enclosed register.

f	Mnemonic	Instruction	Description
0	ADA, b, m	Add to A	$(A) + (M) \rightarrow A$
1	SBA, b, m	Subtract from A	$(A) - (M) \rightarrow A$
2	LDA, b, m	Load A	$(M) \rightarrow A$
3	STA, b, m	Store A	$(A) \rightarrow M$
4	MFA, b, m	Multiply A	$(A) \times (M) \rightarrow AQ$

Note: After multiply is executed $K = 7777_8$ and product has following form



Type 2 Instructions

The operand for Type 2 instructions is defined by y, a 7-bit quantity whose sign is extended to 24 bits for enter A, and extended to 12 bits for enter B_1 and relative jump instructions.

f	s	Mnemonic	Instruction	Description
5	0	RJL, y	Relative Jump to a Lower	$(P) + y \rightarrow P_L$
5	1	RJU, y	Relative Jump to an Upper	$(P) + y \rightarrow P_U$
5	2	ENA, y	Enter A	$y \rightarrow A$
5	3	ENB, y	Enter B_1	$y \rightarrow B_1$

Note on Jumps: The normal sequence of operation is

Execute upper instruction

Execute lower instruction

Advance P (program address register)

where P contains the address of the current instruction pair. This sequence is changed upon execution of a jump instruction as follows:

a) RJU instruction

   Upper/lower F/F set to lower

   $(P) + y \rightarrow P$

   $(P) + 1 \rightarrow P$

   Execute next instruction (at an upper)

b) RJL instruction

   Upper/lower F/F set to upper

   $(P) + y \rightarrow P$

   Execute next instruction (at a lower)

Type 3 Instructions

Three subgroups comprise the Type 3 instructions. They are two register operations, one register operations, and the input/output instructions.

The two register operations function with the register file and are double address transfer operations coupled with arithmetic operations.

TWO REGISTER OPERATIONS

f	s	Mnemonic	Instruction	Description
6	0	ADD i, j	Add	$(R^i) + (R^j) \rightarrow R^j$

TWO REGISTER OPERATIONS
(continued)

f	s	Mnemonic	Instruction	Description
6	1	SUB i, j	Subtract	$(R^j) - (R^i) \rightarrow R^j$
6	2	MSK i, j	Mask	$(R^i) \cdot (\bar{R}^j) \rightarrow R^j$

$j \diagdown i$	0	1
0	0	1
1	0	0

f	s	Mnemonic	Instruction	Description
6	3	IOR i, j	Inclusive or	$(R^i) \overset{\text{logical}}{+} (R^j) \rightarrow R^j$

$j \diagdown i$	0	1
0	0	1
1	1	1

f	s	Mnemonic	Instruction	Description
6	4	SRJ i, j	Subroutine jump	$(R^i_L) \rightarrow R^j_L$ $(R^j_U) \rightarrow R^i_L$
6	5	AWC i, j	Add with carry	$(R^i) + (R^j) +$ overflow $\rightarrow R^j$
6	6	TRF i, j	Transfer	$(R^i) \rightarrow R^j$
7	0	XUU i, j	Upper to upper	$(R^i_U) \rightarrow R^j_U$
7	1	XUL i, j	Upper to lower	$(R^i_U) \rightarrow R^j_L$
7	2	XLU i, j	Lower to upper	$(R^i_L) \rightarrow R^j_U$
7	3	XLL i, j	Lower to lower	$(R^i_L) \rightarrow R^j_L$
7	6	RS1 i, j	Right shift	$(R^i) \rightarrow R^j$ (shifted right 1) (sign bit extended)

Any register operation which uses the program address register as a destination will not alter the normal sequence in regard to the next instruction executed. If the operation occurs at an upper, the next instruction executed will be at a lower. If the operation occurs at a lower, the next instruction executed at an upper after program address register has been advanced by one.

The one register operation instructions function with the register file and are single address operations. Skip instructions function as follows. The normal sequence -- execute upper instruction/execute lower instruction/advance program address register -- is not followed when a skip exit is performed. One instruction is always skipped. A skip exit taken from an upper instruction transfers control to the next sequential upper instruction. A skip exit taken from a lower instruction transfers control to the next sequential lower instruction.

ONE REGISTER OPERATIONS

f	s	i	Mnemonic	Instruction	Description
6	7	0	INC j	Increment	$(R^j) + 1 \rightarrow R^j$
6	7	1	DNG j	Spare	$(R^j) \rightarrow R^j$
6	7	2	ST1 j	Set LSB	$2^{-23} \rightarrow R^j$
6	7	3	CLR j	Clear	$0 \rightarrow R^j$
6	7	4	DNG j	Spare	$(R^j) \rightarrow R^j$
6	7	5	DEC j	Decrement	$(R^j) - 1 \rightarrow R^j$

ONE REGISTER OPERATIONS
(continued)

f	s	i	Mnemonic	Instruction	Description
6	7	6	CMP j	Two's Complement	$-(R^j) \rightarrow R^j$
6	7	7	NOT j	One's Complement	$(\overline{R}^j) \rightarrow R^j$
7	7	0	DSU j	Decrement skip upper	Test (R_U^j) lower 8 bits then $(R_U^j) - 1 \rightarrow R_U^j$ if test = 0, skip next instruction $(R_U^j) = 7777_8$ if test $\neq$ 0, take next instruction.
7	7	1	DSL j	Decrement skip lower	Test (R_L^j) lower 8 bits then $(R_L^j) - 1 \rightarrow R_L^j$ if test = 0, skip next instruction. $(R_L^j) = 7777_8$ if test $\neq$ 0, take next instruction.
7	7	2	ISU, j	Increment skip upper	Test (R_U^j) lower 8 bits then $(R_U^j) + 1 \rightarrow R_U^j$ if test = 0, skip next instruction $(R_U^j) = 0001_8$ if test $\neq$ 0, take next instruction.

ONE REGISTER OPERATIONS
(continued)

f	s	i	Mnemonic	Instruction	Description
7	7	3	ISL, j	Increment skip lower	Test (R^j_L) lower 8 bits then $(R^j_L) + 1 \rightarrow R^j_L$ if test = 0, skip next instruction. $(R^j_L) = 0001_8$ if test ≠ 0, take next instruction.

7	7	4	ZSK, j	Zero skip	Test (R^j)
					if test = 0, skip next instruction
					if test ≠ 0, take next instruction.

7	7	5	NZS, j	Non-zero skip	Test (R^j)
					if test ≠ 0, skip next instruction
					if test = 0, take next instruction.

7	7	6	NSK, j	Negative skip	Test (R^j) sign bit
					if test ≠ 0, skip next instruction
					if test = 0, take next instruction.

7	7	7	PSK, j	Positive skip	Test (R^j) sign bit
					if test = 0, skip next instruction
					if test ≠ 0, take next instruction.

The input/output instructions specify channel by the j-bits and a register file address with i-bits.

INPUT/OUTPUT OPERATIONS

f	s	Mnemonic	Instructions	Description
7	4	OUT, i, j	Output	$(R^i) \rightarrow$ channel j
7	5	INP, i, j	Input	(Channel j) $\rightarrow R^i$

Carry Flip Flop

The carry flip flop is a one bit register used primarily as a carry (or overflow) from a 12-bit arithmetic operation. Such use is that of the carry bit from an add where the lower 12 bits of the two operands are added, then the upper 12 bits and the carry bit are added. It forms the final result of the 24-bit add. However, all of the instructions effect the contents of the flip flop in the following manner:

a) the following instructions clear the carry F/F:
LDA, STA, RJL, RJU, ENA, ENB, MSK, IOR, SRJ, TRF, ST1, CLR, NOT, XUU, XUL, XLU, XLL, INP, OUT, RS1, DSU, DSL, ISU, ISL, ZSK, NZS, NSK, PSK

b) the following instructions put the overflow from an arithmetic operation into the carry F/F:
ADA, SBA, ADD, SUB, INC, DEC

c) the carry F/F is cleared by the CMP instruction except when the two's complement of zero is done, in which case the carry F/F is set of 1

d) the MFA instruction affects the carry F/F in the following manner:

multiplicand \ multiplier	-	+
-	0	0
+	1	0

e) the AWC instruction is the only instruction which has access to the carry F/F via a programmable instruction. The sequence is as follows:

add $(R^i) + (R^j) + $ (carry F/F) store result in R^j

the overflow from this add operation is inserted into the carry F/F.

C. Input/Output

The 449 computer has provision for 8 input channels and 8 output channels. Each can be up to 24 bits in length. In addition, any number of priority interrupts canbe handled with very little additional hardware. In this section inputs, outputs, and interrupts will be discussed and a typical I/O system (that present in the original 449 system) will be shown.

1. Inputs

An input instruction will store 24 bits of data in a register file word specified within the instruction. However, the data is taken into the computer on 12 lines in two 12-bit parallel transfers. When an input instruction is programmed the computer will first store 12 bits of input data in the upper half of the specified register file word. Eight microseconds later it will store 12 bits of input data in the lower half of the specified register file word. An "upper/lower" signal is provided from the computer to tell where the data is being stored. If a 24-bit input is desired the 24 data lines should be combined with the "upper/lower" signal so the proper 12 bits will appear at the computer input lines at the right time.

Any one of eight channels may be specified by three bits in the instruction. These three "channel" lines are provided by the computer. These bits may be used to matrix eight 24-bit channels of data into the computer. The "channel" lines are stable four microseconds before the first 12-bit data input. A one microsecond "input" signal is sent from the computer when 12-bit data is transferred into the computer.

2. Outputs

Output instructions also transfer 24-bit data into two 12-bit transfers. An "upper output" signal is provided when the upper 12 bits of the word specified in the output instruction are available on the output data lines.

Eight microseconds later a "lower output" signal is provided when the lower 12 bits are available on the output data lines. A "clear output" signal is also provided four microseconds before the "upper output" signal.

"Channel" signals are used for outputs as well as inputs. Eight 24-bit registers may be matrixed to the output data lines.

3. Interrupts

An "interrupt request" line is provided. The computer will scan this line after executing the lower instruction in the 24-bit memory word. If a request is present at that time, the computer will send a 26 microsecond pulse on the "interrupt" signal line. Then an interrupt sequence will be executed in place of the next programmed instruction. The contents of P (address 0000, lower) will be placed in K (address 0000, upper). Then the octal number 0410 is stored in P. Thus, the next instruction executed will be that in address 0410 lower.

Notice a multiply instruction programmed within a subroutine will destroy the return address held in K.

Maximum waiting time for an interrupt after request is 56 microseconds if no multiply instructions are programmed. Otherwise, the waiting time is 660 microseconds; or, if two multiply instructions are programmed in one 24-bit address, 1208 microseconds.

An expanded and more flexible interrupt system may be easily added to the 449. In this case the interrupt described above is left unused. The expanded priority interrupt system is based on the use of a different register file for each interrupt. Since these are actually memory locations, the only hardware required is an additional hardware flip flop representing the number of interrupts desired, i.e. 3 interrupts-2 flip flops, 15 interrupts-4 flip flops. These flip flops are additions to the R register. The bits are exclusively under the control of the interrupt system. An interrupt will cause these bits to be changed after an instruction is executed. The address of the next instruction pair is determined by the P register (0000, lower) in the new register file.

In addition to the extension of the R register, an interrupt register is required, consisting of one flip flop for each interrupt. When an interrupt is requested, the associated flip flop is set by an external line. After the interrupt program is completed, the computer will clear the flip flop through an output instruction.

Priority problems occur when more than one interrupt is requested at once. The lines are ordered so the bit to the left will have the highest priority. A priority-decoder network actually generates a number corresponding to the position of the left most bit set in the interrupt register. Other bits to the right of this are ignored. Thus, if two or more interrupt requests are present, only the left most (highest priority) one will be recognized. This means a lower priority interrupt must wait until a higher priority program is completed and the associated bit in the interrupt register cleared.

D. 449 System I/O

A block diagram of the I/O installed in the first 449 computer is shown in Figure 9. The I/O in this application was almost entirely within the computer box, consisting of the keyboard, wheels and the real time clock. In addition, there are 3 bits of input and 3 bits of output associated with the Programmer's Console.

IV. MAINTENANCE CONSOLE

A. General Description

There are two pieces of maintenance equipment associated with the 449 computer--the Memory Loader and Programmer's Console, which is used for loading, program checkout, testing, and demonstration of the computer, and Fault Maintenance Equipment used to debug faults in the computer. When the computer is complete with dust cover, the only communication to the console is through a 3-bit input and 3-bit output channel. If the computer is working properly, this is quite adequate. Since most registers are in memory they can only be viewed under program control. The registers are outputted one octal character at a time to the console for display. If the computer is defective this type of console is not adequate. A defective computer is more easily debugged with a separate piece of equipment which is plugged into the side of the central processor and is called the Fault Maintenance Equipment.

B. Memory Loader and Programmer's Console

The Memory Loader and Programmer's Console consists of a display panel, a memory loader, and a Control Data 350 Perforated Tape Reader. The 449 sits directly over the memory loader and is connected to the console through the single connector on the front of the 449. When the memory is being loaded, the outer dust cover must be removed and three additional connectors inserted.

The console provides the following functions:

a) loads memory from paper tape ,

b) verifies against paper tape ,

c) manual inspection and change of a word of memory ,

d) scans through memory, one word at a time ,

e) steps through program, one word at a time .

The panel for the console is shown in Figure 11. One address and one 24-bit data word are displayed by binary indicators. Push buttons allow for manual entry of data.

If desired, a CRT display can be added and the entire register file displayed. The information would still flow from computer to console, one octal character at a time and the CRT would serve as the memory for viewing all characters at once. The CRT display can be any oscilloscope having conventional x, y, and z inputs. The console can be modified to supply the necessary analog voltages to display the characters in binary or octal form.

A photograph of the engineering model of the Memory Loader and Programmer's Console is shown in Figure 12. The memory loader is on the left and will be sunk below tabletop level. The central console will be sloped for easy viewing.

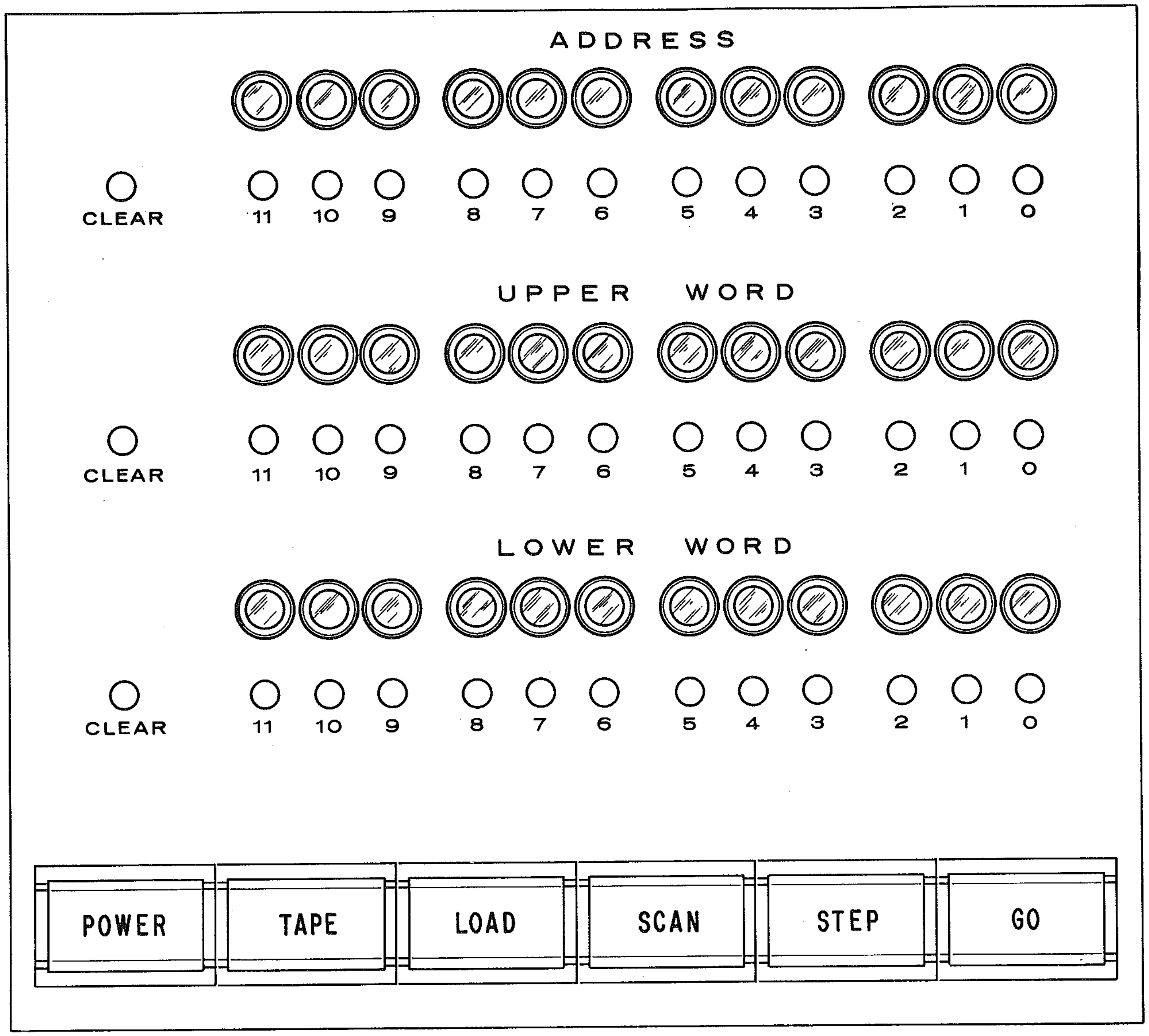


Figure 11: Memory Loader and Programmer's Panel

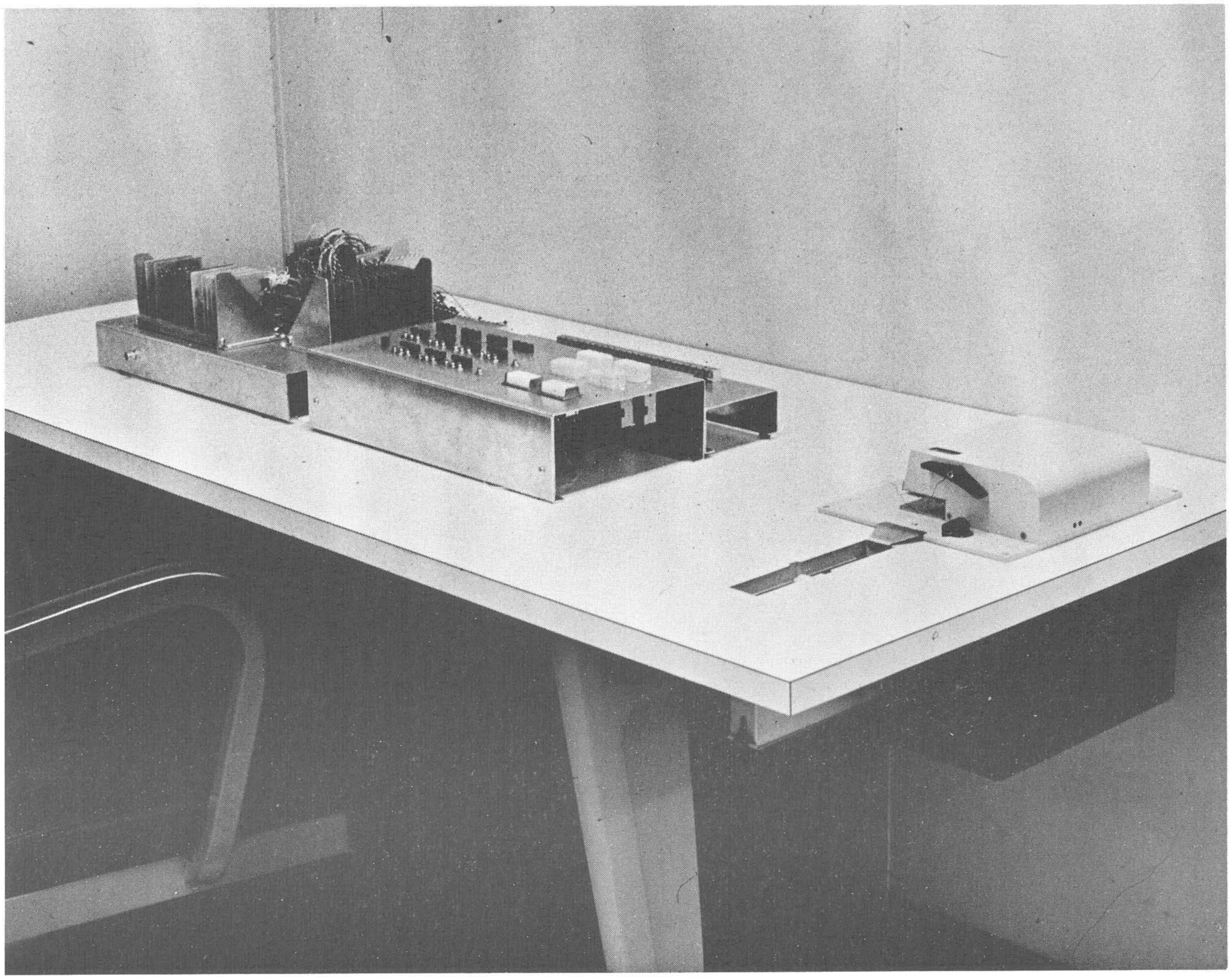


Figure 12: Prototype Memory Loader and Programmer's Console